Dynamic Scheduling and Optimal Reconfiguration of UPF Placement in 5G Networks

Irian Leyva-Pupo
irian.leyva@entel.upc.edu
Universitat Politècnica de Catalunya

Cristina Cervelló-Pastor
cristina@entel.upc.edu
Universitat Politècnica de Catalunya

Christos Anagnostopoulos
christos.anagnostopoulos@glasgow.ac.uk
University of Glasgow

Dimitrios P. Pezaros
dimitrios.pezaros@glasgow.ac.uk
University of Glasgow

ABSTRACT

Multi-access Edge Computing (MEC) is a key technology in the road to 5G and beyond networks. Significant reductions in both latency and backhaul traffic can be achieved by placing server applications, and network functions at the network edge. However, this implies new challenges for their dynamic placement and management. In this paper, we tackle the problem of dynamic placement reconfiguration of 5G User Plane Functions (UPFs) in a MEC ecosystem to adapt to changes in user locations while ensuring QoS and network operator expenditures reduction. In this vein, an Integer Linear Programming (ILP) solution is proposed to determine the optimal UPF placement configuration (e.g., number of UPFs and user-UPF mapping) by considering several cost components along with service requirements. Moreover, a scheduling technique based on Optimal Stopping Theory (OST) is presented to decide the optimal reconfiguration time according to instantaneous values of latency violations and established QoS thresholds. Extensive simulation results demonstrate their effectiveness, achieving significant improvements in metrics such as number of re-computation events, reconfiguration costs, and number of latency violations over time.

CCS CONCEPTS

• **Networks → Network performance modeling**.

KEYWORDS

5G, Multi-access Edge Computing, User Plane Function, Integer Linear Programming, Optimal Stopping Theory

1 INTRODUCTION

The Fifth Generation (5G) of mobile networks promises a fully mobile and connected society with a wide variety of new services and use cases [3]. These services have stringent requirements in terms of latency, reliability, connectivity density, bandwidth, and energy efficiency. The latter imposes radical transformations not only on network architecture but also on its management and orchestration. In this context, technologies such as Software Defined Networking (SDN), Network Function Virtualization (NFV), and Multi-access Edge Computing (MEC) have been defined as key 5G enablers [5].

SDN provides control and user plane separation (CUPS), enabling programmable, flexible, and scalable architectures, whereas NFV reduces capital and operational expenditure and improves management capabilities and resource utilization. Additionally, MEC reduces the end-to-end (E2E) service delay and backhaul traffic, which makes it appealing for the deployment of applications (e.g., virtual/augmented reality and autonomous cars) and User Plane Functions (UPFs) [12]. However, the placement and management of UPFs in the MEC ecosystem is challenging mainly due to user mobility, MEC servers' limited resources, and 5G strict service requirements. Besides, when optimizing UPF placement, we usually face conflicting objectives, such as service latency optimization and session relocation avoidance. As users move, their network response times may increase, implying not only Quality of Service (QoS) degradation but also higher routing costs for network operators. Under these circumstances, frequent and dynamic placement readjustments may be necessary to cope with user mobility while ensuring QoS. Nevertheless, this may produce extra delays in the session data path and service interruptions due to session relocations during placement reevaluations.

In this context, the design of strategies to optimize UPF placement in MEC environments and determining the optimal reconfiguration time becomes crucial. To this aim, the key contributions of this paper can be summarized as follows: (i) a multi-objective Integer Linear Programming (ILP) model for the reconfiguration of UPF placement; (ii) a scheduling technique, based on Optimal Stopping Theory (OST) [8], for the dynamic orchestration of UPF placement according to instantaneous values of latency violations; (iii) a thorough evaluation of the proposed solutions to show their efficiencies. The obtained results demonstrate that when instantaneous values of latency violations are considered along with proper optimization objectives for the UPF placement reconfiguration, significant performance improvements in both the number of re-computation events and QoS can be achieved.

The remainder of this paper is organized as follows. Section 2 presents a literature review related to the fields of UPF placement, dynamic Virtual Network Function (VNF) placement, and OST. In Sections 3 and 4, the proposed solutions for the optimal placement of the UPFs and their dynamic scheduling reconfiguration are provided. Subsequently, the performance evaluation results are analyzed in Section 5. Finally, Section 6 concludes our work.

2 STATE OF THE ART

The 5G UPFs represent the evolution of traditional Serving and Packet Gateways (SGWs and PGWs) from EPC to 5G networks under the CUPS concept. The placement of these network functions has been addressed in a wide variety of research works [13, 14, 16–18]. In [16], the effects of centralized and distributed SGW placement strategies on the backhaul bandwidth are studied. As a result,

the authors conclude that a distributed SGW placement where each base station is co-located with an SGW performs similarly to an optimized SGW placement. Taleb et al., in [18], address the joint placement of SGW and PGW by considering user mobility patterns, service delay, and relocation constraints. Their main objective is to minimize the user plane response time as well as SGW relocations. In [13] and [14], the UPF placement in a native 5G architecture is addressed. These two works conceive various placement strategies target at reducing deployment and operational expenditures while satisfying 5G service requirements. Nonetheless, all the works mentioned above are based on static approaches making the proposed strategies unable to adapt to network variations (e.g., user mobility and traffic).

Peters et al. in [17] propose a learning approach to proactively take session management actions based on user mobility and activity predictions. The latter, along with Protocol Data Unit (PDU) session requirements, is used to make decisions regarding the insertion or not of intermediate UPFs in the users' data path and select the best candidate for their placement. In [7], a VNF service replication strategy to respond to users' requests in a MEC ecosystem proactively is presented. To this aim, two ILP models are proposed to cope with two conflicting objectives: enhancing Quality of Experience (QoE) during service migration and reducing resource consumption and deployment costs. However, these approaches imply a waste of resources in MEC servers by reserving resources that might never be used. Moreover, they are based on individual users' behavior, which means that they need to be executed every time a user changes the edge node (EN).

Cziva et al. in [6] propose an ILP model to minimize the E2E service latency along with a dynamic placement scheduler to forecast when the placement needs to be reconfigured. Their main objective is to guarantee the established QoS levels, whereas frequent placement recalculations are avoided. To determine the optimal reconfiguration time, they rely on OST. The OST has been widely adopted to solve optimization problems [4, 11, 19] due to its effectiveness. In [4], the authors propose a model that adopts OST principles to decide when the optimal time is to take mitigation actions in ENs (i.e., upgrade the current services or offload tasks). In this way, the ENs can adapt their configuration to ensure the desired QoS. Additionally, in [11], the main aim is to minimize energy consumption during dynamic service migration in a MEC environment. In this vein, OST is applied to obtain the optimal migration energy expectation and to select the target migration node. Similarly, Wu et al. in [19] use OST to choose the best nodes for the cache placement so that energy saving is maximized.

From the aforementioned studies, the closest one to ours is [6]. Our work and [6] are similar in the sense that both propose ILP solutions for the placement of VNF in MEC, which contemplate E2E user plane latency. Moreover, both works rely on OST to determine the optimal time to reevaluate the VNF placement w.r.t. the maximum number of latency violations allowed. However, unlike [6], our proposed model seeks to diminish the effects of placement reevaluation by taking into account the current placement conditions. Furthermore, instead of a one to one user-VNF mapping, we consider that a UPF instance can be shared by several users as long as its capacity is not exceeded. Besides, contrary to [6], where the reconfiguration decision is made regarding the cumulative number of latency violations over time, the present paper triggers the reconfiguration events based on its instantaneous values.

3 OPTIMAL UPF PLACEMENT RECONFIGURATION

In this section, the network model and used notation are presented. Afterward, the formulation of the UPF Placement Reconfiguration (UPR) problem is introduced. The UPR problem is formulated as a multi-objective optimization problem (MOOP). It is intended to find the optimal location and number of UPFs as well as the best assignment of users to reduce expenditures while satisfying users' service requirements.

3.1 Network Model and Notation

Figure 1 depicts a simplified view of the 5G architecture. The 3GPP defined this architecture [1, 2] based on the principles of CUPS, network slicing, and service-based architecture. The 5G user plane is compound by the UPFs, which are responsible for processing data plane packets between the Radio Access Network (RAN) and the Data Network (DN), QoS handling, packet routing and forwarding, lawful interception, etc. To perform all these functionalities, the UPFs rely on the Session Management Functions (SMFs), located in the control plane. The SMFs are in charge of selecting, controlling, and managing the UPFs to establish PDU sessions.

The 5G network is represented as a graph G(N, E, U), where N, E, and U denote the set of network nodes, the links between them and users with active PDU sessions, respectively. The set of network nodes is formed by UPF potential locations (N_c), already deployed UPFs (N_u), aggregation points (APs, (N_a)) and access nodes (N_r) whereas the set of users is extended to the set of active sessions (N_s). The PDU sessions are characterized by the following parameters: E2E service latency requirement (L^s_{ser}), computing resource demand (D^s) -e.g., CPU and RAM- and the minimum number of anchor UPFs (K^s_u) to guarantee the service reliability. For simplicity, we assume that there are no capacity limitations in the bandwidth of the links.

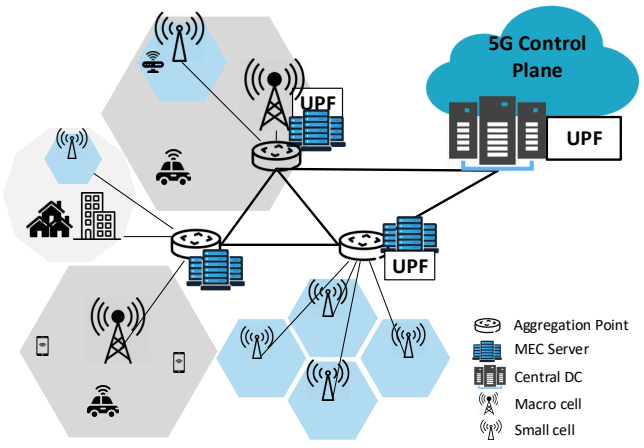


Figure 1: Deployment of 5G UPFs in a MEC ecosystem.

Table 1: Sets, parameters and variables notation

Notation	Description
	Sets
N_c	Set of UPF potential locations (e.g., MEC servers)
N_u	Set of UPFs already deployed
N_a	Set of aggregation points
N_r	Set of access nodes
N_s	Set of PDU session requests
	Parameters
L^s_{ser}	Service latency requirement of PDU session $s \in N_s$
L^s_{ra}	Delay in the link between the access node $r \in N_r$ of the PDU session $s \in N_s$ and its aggregation point $a \in N_a$
L^s_{ac}	Delay in the link between aggregation point $a \in N_a$ of the PDU session $s \in N_s$ and a candidate location $c \in N_c$
T^s_{proc}	Processing time in the data path of PDU session $s \in N_s$
T^s_{prop}	Propagation time in the data path of PDU session $s \in N_s$
K^s_u	Number of UPFs required for PDU session $s \in N_s$
D^s	Computing resources required by the PDU session $s \in N_s$
C_c	Hardware capacity (e.g., CPU and RAM) at location $c \in N_c$
F^d_c	Cost of installing a UPF in location $c \in N_c$
F^o_c	Cost of running a UPF in location $c \in N_c$
F_{ac}	Cost of routing associated to link ac, $a \in N_a$ and $c \in N_c$
$F^m_{c'c}$	Cost of migrating a UPF from location c' to $c \in N_c$
F^r_s	Cost of reassigning a PDU session $s \in N_s$
	Indicators
P^s_c	1 if PDU session $s \in N_s$ was assigned to a UPF in $c \in N_c$ before the placement reconfiguration
X_c	1 if there was a UPF placed in $c \in N_c$ before reconfiguring
V^u_c	1 if UPF $u \in N_u$ was deployed in location $c \in N_c$
	Binary variables
x_c	1 if there is a UPF in location $c \in N_c$
h_c	1 if it has been a change in location $c \in N_c$
n_c	1 if a new UPF is deployed in location $c \in N_c$
δ_c	1 if there is a UPF $\in N_u$ deployed in location $c \in N_c$
v^u_c	1 if UPF $u \in N_u$ is deployed in location $c \in N_c$
$m^u_{c'c}$	1 if UPF $u \in N_u$ located in c' is migrated to $c \in N_c$
p^s_c	1 if PDU session $s \in N_s$ is assigned to a UPF in $c \in N_c$

Table 1 summarizes the notation used for the formulation of the UPR problem.

3.2 Optimal UPF Placement Reconfiguration

The UPR problem seeks to minimize capital and operational expenditures produced as a result of reconfiguration events. To this aim, the following cost components are considered:

- Deployment Cost (C_{dep}): It includes the costs related to the installation of new UPF instances.

$$C_{dep} = \sum_{c \in N_c} F^d_c \cdot n_c \tag{1}$$

- Running Cost (C_{run}): It deals with the cost of operating UPF instances, and it is expressed in terms of the number of deployed UPFs.

$$C_{run} = \sum_{c \in N_c} F^o_c \cdot x_c \tag{2}$$

- Routing Cost (C_{rou}): It expresses the cost of routing PDU sessions from their APs to their assigned UPFs (L_{ac}). Note that by reducing this cost, network response time can also be improved, as it is expressed in terms of propagation delay.

$$C_{rou} = \sum_{c \in N_c} \sum_{a \in N_a} \sum_{s \in N_s} F_{ac} \cdot L^s_{ac} \cdot p^s_c \tag{3}$$

- Migration Cost (C_{mig}): It represents the cost of migrating a UPF instance from one location to another. It is computed regarding the UPFs that were instantiated as a result of the previous placement (N_u).

$$C_{mig} = \sum_{c' \in N_c} \sum_{c \in N_c} \sum_{u \in N_u} F^m_{c'c} \cdot m^u_{c'c} \tag{4}$$

- Reassignment Cost (C_{rea}): It is the cost for reassigning PDU sessions during reconfigurations. It is measured as a penalty that the service provider (SP) has to pay for interrupting a session. The reassignment of a PDU session is indicated by $[p^s_c - P^s_c]^+$ where $[f(x)]^+ = max\{f(x), 0\}$. Specifically, this expression is 1 if a session is assigned to a UPF different from the one that it had prior to the reconfiguration.

$$C_{rea} = \sum_{c \in N_c} \sum_{s \in N_s} F^r_s \cdot [p^s_c - P^s_c]^+ \tag{5}$$

The main objective of the UPR problem, see (6), is to minimize the effects of the above cost components during the placement reevaluation. Since the solution to this problem results in the optimization of conflicting objectives, a trade-off among them has to be found. To this end, these components should be normalized and added together. Additionally, weight factors (α_i) can be included to specify their relative importance. It should be noted that this model can also be used for initial or static UPF placement by removing the terms that depend on previous time instances (i.e., migration and reassignment costs) and setting the indicators P^s_c, X_c and V^u_c at zero. The UPR problem can be formulated as follows:

$$\text{Min } \alpha_1 \cdot C_{dep} + \alpha_2 \cdot C_{run} + \alpha_3 \cdot C_{rou} + \alpha_4 \cdot C_{mig} + \alpha_5 \cdot C_{rea} \tag{6}$$

s.t.:

$$p^s_c \leq x_c \qquad \forall s \in N_s, \forall c \in N_c \tag{7}$$

$$x_c \leq \sum_{s \in N_s} p^s_c \qquad \forall c \in N_c \tag{8}$$

$$\sum_{c \in N_c} p^s_c \geq K^s_u \qquad \forall s \in N_s \tag{9}$$

$$\sum_{s \in N_s} D^s \cdot p^s_c \leq C_c \qquad \forall c \in N_c \tag{10}$$

$$T^s_{proc} + T^s_{prop} \leq L^s_{serv} \qquad \forall s \in N_s, \forall c \in N_c \tag{11}$$

$$m^u_{c'c} = v^u_c \wedge V^u_{c'} \qquad \forall u \in N_u, \forall c', c \in N_c, c' \neq c \tag{12}$$

$$\sum_{c' \in N_c} \sum_{c \in N_c} m^u_{c'c} \leq 1 \qquad \forall u \in N_u \tag{13}$$

$$\sum_{c \in N_c} v^u_c \leq 1 \qquad \forall u \in N_u \tag{14}$$

$$\sum_{c \in N_c} \sum_{u \in N_u} v^u_c \leq |N_u| \tag{15}$$

$$x_c = 1 \Rightarrow n_c \oplus \sum_{u \in N_u} v^u_c = 1 \qquad \forall c \in N_c \tag{16}$$

$$h_c = n_c \oplus \sum_{c' \in N_c} \sum_{u \in N_u} m^u_{c'c} \qquad \forall c \in N_c \tag{17}$$

$$n_c + \sum_{c' \in N_c} \sum_{u \in N_u} m^u_{c'c} \leq 1 \qquad \forall c \in N_c \tag{18}$$

$$x_c, h_c, n_c, v^u_c, m^u_{c'c}, p^s_c \text{ binary} \qquad \forall s \in N_s, \forall c \in N_c \tag{19}$$

The constraint (7) specifies that a PDU session cannot be assigned to a candidate unless there is a UPF at that location, whereas (8) avoids the deployment of empty UPFs. Additionally, inequality (9) ensures that each PDU session is served by the minimum number of UPFs needed to meet its reliability requirement. Constraint (10) guarantees that the physical resources available at a location are not exceeded by the service demands of its deployed UPF instance. We assume that various PDU sessions can share the resources assigned to a VNF instance according to its available capacity.

The inequality (11) ensures that the overall delay in the data plane (Round-Trip-Time, RTT) does not exceed the service latency requirement (L^s_{ser}). The latency of a PDU session is defined in terms of the processing time (T^s_{proc}) of the network elements that form its data path and the propagation delay (T^s_{prop}) between them, see (20) and (21). Where T_r, T_a, T_u and T_d represent the processing time of the access nodes, APs, UPFs and DN whereas L^s_{ra}, L^s_{ac} and L^s_{cd} are the propagation segments between them. The application servers and the UPFs are assumed to be co-located in the ENs ($L^s_{cd} = 0$).

$$T^s_{proc} = 2 \cdot (T^s_r + T^s_a + T_u \cdot p^s_c + T_d) \tag{20}$$

$$T^s_{prop} = 2 \cdot (L^s_{ra} + L^s_{ac} \cdot p^s_c + L^s_{cd}) \tag{21}$$

The migration of a UPF instance from a source location c' to a target position c is indicated by constraint (12). It is determined upon the UPFs that were instantiated during the previous placement event. This constraint is nonlinear but can be expressed in a linear form as follows:

$$m^u_{c'c} \leq v^u_c \qquad \forall u \in N_u, \forall c', c \in N_c, c' \neq c$$

$$m^u_{c'c} \leq V^u_{c'} \qquad \forall u \in N_u, \forall c', c \in N_c, c' \neq c$$

$$m^u_{c'c} \geq v^u_c + V^u_{c'} - 1 \qquad \forall u \in N_u, \forall c', c \in N_c, c' \neq c$$

Expression (13) stipulates that a UPF instance can be migrated to one location at most during the placement reevaluation. In addition, inequality (14) forces the assignment of a UPF instance already deployed ($u \in N_u$) to just one candidate location. Moreover, constraint (15) specifies that the size of this set, at the end of a reconfiguration, cannot be greater than before it. In other words, after a placement reevaluation, the number of UPFs that belongs to the set of UPFs already deployed can remain constant or decrease, but it cannot increase.

Inequality (16) indicates that if after a reconfiguration, there is a UPF at a candidate location; this is either because of the deployment of a new UPF instance or the presence of an already deployed UPF. The latter may be due to a migration or simply because there was not change in its location. This expression is nonlinear and can be linearized by introducing a new binary variable (δ_c, where $\delta_c = 1 \leftrightarrow \sum_{u \in N_u} v^u_c = 1$) and adding the following constraints:

$$n_c + \delta_c \leq 2 - X_c \qquad \forall c \in N_c$$

$$n_c + \delta_c \geq X_c \qquad \forall c \in N_c$$

$$\sum_{u \in N_u} v^u_c \geq \delta_c \qquad \forall c \in N_c$$

$$\sum_{u \in N_u} v^u_c \leq |N_u| * \delta_c \qquad \forall c \in N_c$$

Constraint (17) expresses that if there is a change in a candidate location (h_c), this is because either a new UPF was deployed or an existent one was migrated to it. A change in a location is determined by comparing its current state with the previous one, in terms of deployed UPFs, $h_c = [x_c - X_c]^+$. Please, note that h_c only considers changes that have a negative impact on the overall cost, thus omitting changes caused by the removal of UPF instances. Since constraint (17) is nonlinear, it can be replaced by the following expressions:

$$h_c \leq \sum_{c' \in N_c} \sum_{u \in N_u} m^u_{c'c} + n_c \qquad \forall c \in N_c$$

$$h_c \geq \sum_{c' \in N_c} \sum_{u \in N_u} m^u_{c'c} - n_c \qquad \forall c \in N_c$$

$$h_c \geq n_c - \sum_{c' \in N_c} \sum_{u \in N_u} m^u_{c'c} \qquad \forall c \in N_c$$

$$h_c \leq 2 - n_c - \sum_{c' \in N_c} \sum_{u \in N_u} m^u_{c'c} \qquad \forall c \in N_c$$

Moreover, constraint (18) restricts the type of change in a location; mainly, it can be due to a new deployment or a migration, but not for both reasons. Finally, constraint (19) represents the binary nature of the variables used in the formulation.

4 DYNAMIC SCHEDULING FOR THE UPR

This section introduces a mechanism called Skeptical Scheduling Reconfiguration (SSR) for the dynamic orchestration of the UPF placement. Firstly, the SSR problem is formulated as an optimal stopping problem (OSP). Subsequently, its optimal stopping rule is provided along with the fundamentals and principles adopted from OST that prove its optimality.

4.1 Skeptical Scheduling Reconfiguration Problem

Given a UPF placement, product of either an initial deployment or a reconfiguration, in which all the PDU session requests were assigned to at least one UPF according to their service requirements, we need to consider some variations in their propagation delay over time due to user mobility. These variations may cause QoS degradation when the distance between the users and their assigned UPFs increases. Namely, a service latency violation is produced when the user plane response time exceeds the service latency requirement. Let's define $I^s_t \in [0, 1]$ as a random variable (r.v.), indicating whether the service latency of a PDU session s is affected or not at time t.

$$I^s_t = \begin{cases} 1 & \text{if the service latency of session } s \text{ is violated at } t \\ 0 & \text{otherwise} \end{cases}$$

Hence, the overall number of sessions with latency violations at a given time t (L_t) can be defined as follows:

$$L_t = \sum_{s \in N_s} I^s_t \tag{22}$$

When a UPF placement is no longer optimal ($L_t \neq 0$), its readjustment is necessary to reestablish the system's QoS levels. However, placement readjustments are resource consuming and may imply additional delays in the user plane, service interruption, and extra costs. Since these events may involve changes in the number of UPFs (addition or removal), in their locations (migration) as well

as reassignment of PDU sessions. Consequently, unnecessary and frequent UPF placement re-computations must be avoided as much as possible and only triggered when needed. Hence, we deal with the problem of determining when the optimal time is to initiate a UPR so that its negative effects are minimized.

Let us assume that at each time instance t, the system can tolerate a maximum number of sessions with latency violations ($\theta > 0$) without requiring the activation of a re-computation event and thus avoiding to incur additional costs and affect other users. However, if this threshold is exceeded, a UPF placement recalculation is required, and an expected cost ($\mathbb{E}[R]$) is incurred. The latter is expressed as a function of the expected number of affected sessions, see (23). The SPs can define the threshold θ according to their service level agreement, e.g., type of service or subscribers profile.

$$\mathbb{E}[R] = \sum_{c \in N_c} \sum_{s \in N_s} [p_c^s - P_c^s]^+ \cdot P(p_c^s \neq P_c^s) \tag{23}$$

The main objective of the SSR problem is to forecast when the system is about to exceed the established threshold to activate the placement reconfiguration in advance and diminish its repercussions in the overall system. Specifically, to tolerate as many latency violations as possible at each time t without exceeding θ, to delay or even avoid placement re-computations. If the number of sessions with latency violations is above the established threshold, an expected reconfiguration cost ($\mathbb{E}[R]$) is estimated. Thus, the reward function of the SSR approach w.r.t. a maximum number of allowed latency violations can be defined as follows:

$$Y_t(L_t) = \begin{cases} L_t & \text{if } L_t \leq \theta \\ -\beta\,\mathbb{E}[R] & \text{if } L_t > \theta \end{cases} \tag{24}$$

where β is a weight factor to adjust the importance of the reconfiguration cost to the reward function.

The target is to determine the optimal time t^* when it is worthy to stop observing the parameter L_t and proceed to readjust the UPF placement. In other words, find the stopping rule that maximizes the expected reward function in (24).

Problem 1. *Given a sequence of events defined by L_t, a maximum QoS threshold θ and an expected reconfiguration cost $\mathbb{E}[R_t]$, seek the optimal decision epoch t^* where the supremum of Y_t is attained:*

$$\sup_{t \geq 0} \mathbb{E}[Y_t(L_t)] \tag{25}$$

4.2 Optimal Skeptical Scheduling Reconfiguration

The theory of optimal stopping is concerned with the problem of choosing a time to take a given action based on sequentially observed r.v. to maximize an expected payoff or to minimize an expected cost [8]. The SSR problem belongs to the group of OSPs with infinite horizon where at each time interval or decision epoch t, we must take one of the following decisions: (i) continue to the next time slot ($t+1$) and do not reconfigure the placement or, (ii) stop and readjust the placement. OSPs are defined by a sequence of observations (r.v.) $X_1, X_2, \ldots, X_t$ whose joint distribution is assumed to be known and a sequence of reward/cost functions $Y_1, Y_2, \ldots, Y_t$, where $Y_t = y_t(x_1, x_2, \ldots, x_t)$.

One approach widely used to solve OSPs due to its simplicity is the One-Stage Look Ahead (1-SLA) rule.

Definition 4.1. For stopping problems, the 1-SLA rule is described by the stopping time

$$t^* = \inf\{t \geq 0 : Y_t \geq \mathbb{E}[Y_{t+1} | \mathbb{F}_t]\} \tag{26}$$

where $\mathbb{F}_t$ is the σ-fields generated by the observations $X_1, \ldots, X_t$. Namely, it represents our knowledge of the r.v. X_t up to time t.

The 1-SLA rule indicates at each decision epoch t whether to stop or continue according to the expected value of the reward function in the next stage, $t+1$. Specifically, it calls for stopping at the first time t for which the reward Y_t for doing it is at least as good (high) as the expected reward for continuing to the next stage and then stopping. An essential condition for the optimality of the 1-SLA rule calls for stopping is the monotonicity of the problem.

Definition 4.2. Let A_t denote the event $\{Y_t \geq \mathbb{E}(Y_{t+1} | \mathbb{F}_t)\}$. The stopping rule problem is monotone if $A_0 \subset A_1 \subset A_2 \ldots$

In other words, if the 1-SLA rule calls for stopping at stage t due to event A_t, then it will also call for stopping at all the future stages (e.g., $t+1, t+2, \ldots$) regardless of the value of the future observations, since $A_t \subset A_{t+1} \subset A_{t+2} \ldots$

Theorem 4.3. *In monotone stopping rule problems, the 1-SLA rule is optimal.*

Proof. Refer to [8]. □

To solve the SSR problem in (25), we derive an optimal stopping rule based on the 1-SLA rule and prove its optimality.

Theorem 4.4. *Given a maximum QoS threshold θ and a sequence of latency violations $L_1, \ldots, L_t$ w.r.t. the last optimal UPF placement ($L_0 = 0$), the optimal stopping time (t^*) for the SSR problem in* (25) *is:*

$$t^* = inf\{t \geq 0 : \sum_{l=0}^{\theta} lP(L = l) - \lambda\,\mathbb{E}[R](1 - \sum_{l=0}^{\theta} P(L = l)) \leq L_t\} \tag{27}$$

Proof. Given that $L_t \leq \theta$, the conditional expectation of Y_{t+1} is given by

$$\begin{aligned} \mathbb{E}[Y_{t+1} | L_t \leq \theta] &= \mathbb{E}[L_{t+1} | L_t \leq \theta, L_{t+1} \leq \theta] P(L_{t+1} \leq \theta) - \\ &\quad \mathbb{E}[\lambda\,\mathbb{E}[R] | L_t \leq \theta, L_{t+1} > \theta] P(L_{t+1} > \theta) \\ &= \mathbb{E}[L_{t+1} | L_{t+1} \leq \theta] P(L_{t+1} \leq \theta) - \\ &\quad \mathbb{E}[\lambda\,\mathbb{E}[R] | L_{t+1} > \theta](1 - P(L_{t+1} \leq \theta)) \\ &= \sum_{l=0}^{\theta} lP(L = l) - \lambda\,\mathbb{E}[R](1 - \sum_{l=0}^{\theta} P(L = l)) \end{aligned}$$

Thus, by comparing the current reward, $Y_t(L_t) = L_t$, with the one expected at the next stage, we obtain that the UPF placement readjustment must be triggered at the first time instance t such that $\mathbb{E}[Y_{t+1} | L_t \leq \theta] \leq L_t$. □

For the 1-SLA to hold optimal to the SSR problem, it is a requirement for the stopping rule proposed in (27) to be monotone.

Theorem 4.5. *In the SSR problem, the 1-SLA is optimal and maximizes the expected reward defined in* (24).

Proof. The SSR problem in (25) is monotone if the difference $\mathbb{E}[Y_{t+1}|L_t \leq \theta] - Y_t(L_t)$ is non-increasing with L_t. This condition is satisfied if the $\mathbb{E}[Y_{t+1}|L_t \leq \theta]$ is non-increasing and $Y_t(L_t)$ is non decreasing almost surely. This can be easily proved, since the left side of inequality (27) remains constant and its right side is increasing over L_t when L_t is below the established threshold ($L_t \leq \theta$). Thus, the 1-SLA rule proposed in (27) is optimal for the SSR problem. □

5 EVALUATION

This section summarizes the simulation results of the proposed solutions. First, aspects related to the simulation setup are presented. Next, we discuss the behavior of the UPR model when various sets of weight factors are considered. Finally, the performance of the SSR mechanism is analyzed in comparison to several baseline schemes.

5.1 Simulation setup

A 5G network scenario composed of 121 access nodes (gNBs) and 13 MEC servers is considered. The gNBs are connected to the ENs through 13 APs. Both APs and ENs are co-located along with the access nodes. The APs are assumed to have a full mesh connection where any AP has a direct link with the others. The inter-site distances are 500 m and 200 m for gNBs located in urban and dense urban areas, respectively. The MEC servers have a processing capacity of 15 CPU and are placed at a maximum distance of 1 km from their assigned gNBs. The initial number of UPFs (N_u=7) and their assigned PDU sessions was obtained through the UPR model by considering deployment, running, and routing costs and giving more importance to the latter (i.e., α_1=0.3, α_2=0.3 and α_3=0.4). For the service demand, 1000 users (vehicles) each with one active PDU session were considered. The PDU sessions have a service latency requirement of 1 *ms* and require just one UPF ($K_u^s = 1$) and 0.1 CPU units to be served. The mobility of the users was generated using the mobility patterns generator CityMob [15] in a realistic downtown model (DM). The user mobility is independent; namely, the mobility of one user does not affect the others. Additional parameters used in the simulation experiments are shown in Table 2.

Table 2: Simulation parameters.

Notation	Description	Value
	UPF Placement	
T_r	RTT delay in the RAN (μs)	500
T_u	Processing time of UPFs (μs)	100
T_{ap}	Processing time of AP (μs)	15
T_d	Processing time of DN (μs)	200
	Propagation delay in optical links ($\mu s/km$)	5
	Number of gNBs per MEC server	[8,10]
	SSR Mechanism	
β	Weight factor of the reconfiguration cost	0.1
	CityMob	
m	Mobility model	3 (DM)
n	Number of users	1000
t	Simulation time (s)	60000
s	Maximum speed of the users (m/s)	40
d	Distance between streets (m)	100
w x d	Dimensions of the grid (km^2)	5x5
a	Number of accidents	0
x,y,X,Y	Downtown limits (km)	1, 1, 2, 2
p	Probability of starting in the downtown	0.45

We modeled the number of sessions with latency violations as a Poisson distribution with a mean of λ=20. To fit this distribution, we observed the number of latency violations at each instance during the simulation time for a UPF placement without reconfiguration and calculated their average values. In real-world scenarios, this parameter can be determined based on SP historical data. All the simulations were performed on a workstation with a 3.30 GHz Intel Core-i9 processor and 64 GB of RAM. For the implementation of the UPR model, the Python-based package Pyomo [10] was selected along with Gurobi [9] as its underlying solver.

5.2 UPF Placement Reconfiguration

To assess the performance of the UPR model, we analyzed the relationships between its cost components as well as their impact on various aspects of the system. Specifically, the following metrics were studied: maximum and mean delays, number of deployed UPFs, maximum number of migrations (Mig.), average imbalance (Imb.), number of reassigned PDU sessions, total number of sessions with latency violations, number of re-computation events (R.E.), and execution time. Table 3 summarizes the results for a simulation period from five hours and a placement reconfiguration based on the SSR solution with an upper bound on the QoS metric of 3% of users with latency violations (θ = 30). These results focus mainly on the variation in the importance of the routing cost (C_{rou}), since the trigger event (L_t) for the placement reevaluation depends directly on its optimization. It is important to note that we do not include all the Pareto-optimal solutions but rather a representative set.

From Table 3, we can appreciate how the mean and maximum propagation delays in the segment AP-UPF (L_{ac}) decrease as the importance of C_{rou} increases. Concretely, these parameters were decremented around two and three times their initial values, respectively, for $\alpha_3 \geq 0.7$. However, this behavior is not only conditioned by the routing cost but also the other terms of the objective function. This can be better appreciated in the experiments with row IDs **d–e**, **f–g**, **h–i**, and **j–k** where we kept the weight factor α_3 constant and varied the importance of other cost components. From these examples, it can be noticed that both the maximum and mean delays have a greater reduction in their average values when the reassignment cost is omitted or has lower importance. Furthermore, by comparing **f** with **e** or **h** with **g**, we should notice that a higher value in the weight factor of a cost component (e.g., α_3) does not necessarily mean an improvement in its performance. We need to consider the effects of the other terms, as well.

On the other hand, an increase in the importance of C_{rou} has a negative impact on the number of deployed and migrated UPFs as well as on the number of reassigned sessions. The latter is remarkable for values of $\alpha_3 \geq 0.5$, where the reevaluation events resulted in either the new deployment or migration of UPF instances. Furthermore, in experiments where new UPFs were deployed (i.e., **k**, **l** and **m**), an increase of more than 30% in the average imbalance was noticed. Concerning the session reassignment metric, its maximum and average values varied from less than 13% of users for $\alpha_3 \leq 0.3$ to more than 60–80% for $\alpha_3 \geq 0.5$. Similarly, the total number of reassigned users increased with the optimization of the latency. This is noticeable when there is no variation in the UPF locations (i.e., experiments **a–e**) since more users need to be relocated to achieve

Table 3: UPR Model: Simulation Results

ID	Weight Components					Metrics											
						Max Delay (μs)	Mean Delay (μs)	No. UPFs	No. Mig.	Imb (%)	No. Relocations			$\sum_t L_t$	No. R.E.	Execution Time (s)	
	α_1	α_2	α_3	α_4	α_5	Aver	Aver	Aver	Max	Aver	Max	Aver	Total	Total	Total	Max	Aver
a	0.2	0.2	0.1	0.3	0.2	18.42	6.43	7	0	0.29	67	48	3763	4878	79	26.79	18.32
b	0.2	0.2	0.2	0.2	0.2	18.32	6.32	7	0	0.28	71	50	4808	4942	96	36.92	20.96
c	0.2	0.2	0.3	0.1	0.2	17.55	5.66	7	0	0.23	124	94	6971	4595	74	46.88	29.27
d	0.2	0.2	0.4	0	0.2	17.58	4.97	7	0	0.30	213	160	9300	4410	58	56.27	31.86
e	0.2	0.2	0.4	0.2	0	12.91	3.06	7	0	0.32	745	716	19343	3683	27	53.72	38.55
f	0.2	0.1	0.5	0.1	0.1	14.23	3.09	7	1	0.31	675	545	11987	2327	22	45.40	19.88
g	0	0.3	0.5	0.2	0	9.75	2.10	7	3	0.28	880	782	3127	2301	4	52.92	33.12
h	0.1	0.1	0.6	0	0.2	12.75	3.54	7	1	0.25	422	422	422	164	1	8.07	8.07
i	0.1	0.2	0.6	0	0.1	10.03	2.15	7	3	0.28	772	634	5704	2296	9	21.55	17.96
j	0.1	0.1	0.7	0	0.1	10.62	2.15	7	3	0.28	772	642	4492	2254	7	9.42	7.56
k	0.1	0.1	0.7	0.1	0	7.91	2.07	8	0	0.73	805	805	805	88	1	10.27	10.27
l	0	0.1	0.8	0.1	0	9.01	1.76	8	2	0.63	842	842	842	88	1	5.23	5.23
m	0	0.1	0.9	0	0	9.01	1.44	9	2	0.66	850	850	850	88	1	5.29	5.29

higher latency reduction. Regarding the total number of latency violations, substantial reductions are achieved by optimizing the routing cost. However, this is at the expense of further transformations during the reconfigurations, either by increasing the number of UPFs or changing their locations.

In general, a decrease in the number of reconfigurations, along with the increasing importance of the routing cost, can be observed in Table 3. This is mainly because more users were assigned to nearer UPFs to reduce the impact of the overall latency in the objective function. Nevertheless, this behavior is not steady, and there are some cases (i.e., rows **b**, **i**, and **j**) where a higher value in α_3 resulted in more reconfigurations. By comparing **j** with **k**, we notice that, sometimes, this increment is due to a variation in the importance of the reassignment cost (α_5). When C_{rea} is considered ($\alpha_5 \geq 0.1$), fewer sessions are reassigned to nearer UPFs, and there is, therefore, a higher probability of experiencing latency violations, which translates into a higher frequency of re-computation events. However, this is not the case for experiments **b** or **i** when they are compared to others with similar weight factors (e.g., **a** or **h**). The main reason for such behavior is that each reconfiguration event is triggered under particular conditions (e.g., user-UPF assignment and number of sessions with latency violations). Besides, they all produce different placement configurations, even when their weight factors are similar. In the end, these slight differences add up, and their effects are reflected in global metrics like the total number of reconfigurations or latency violations. Furthermore, regarding the execution time, its average and maximum values were always less than 60 s for every experiment.

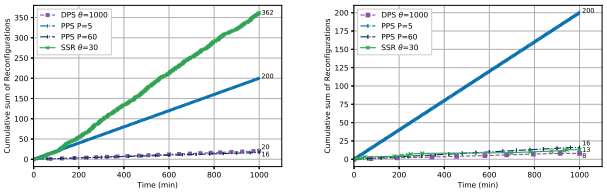
As we can see, the optimization of one or more parameters has a significant impact on the others since we face conflicting objectives. These objectives can be divided into three main subgroups. One is related to the number of UPFs and is formed by the deployment and running costs. Another is associated with the relocation of PDU sessions and comprises the migration and reassignment cost components, and the other is linked to the routing cost. Overall, there is no single best solution when encountering MOOP, but rather a set of multiple optimal solutions (Pareto-Fronts). Thus, the selection of one solution over another depends on what we seek to optimize.

5.3 Dynamic Placement Scheduling

We evaluated the effectiveness of the SSR solution by considering two sets of weights for the UPR model. The first set considers all the terms in the objective function as equally important (all $\alpha_i = 0.2$), whereas the second favors the routing cost over the rest (α_1=α_5=0, α_2=0.3, α_3=0.5 and α_4=0.2). For these sets, we collected samples of the number of latency violations every minute until 1000 samples were acquired. The performance of the proposed scheduling mechanism was analyzed according to the following metrics: number of reconfigurations, number of reassigned sessions, number of UPF migrations, and number of latency violations. To prove its effectiveness, we proceeded to compare it with the following benchmarks:

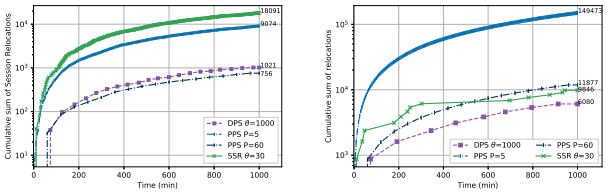
- Periodic Placement Scheduling (PPS): The UPF placement recomputation is executed periodically at fixed time intervals (i.e., every 5 and 60 minutes).
- Dynamic Placement Scheduling (DPS): This strategy adopts the model proposed in [6], where the placement is reevaluated w.r.t. the maximum allowed number of latency violations over time (i.e., a threshold of 1000 latency violations was considered).

5.3.1 Reconfiguration Cost. This cost is analyzed according to the number of re-computations, the number of reassigned sessions, and the number of deployed and migrated UPFs for a UPR based on equally and unequally weighted cost components. In terms of the number of placement readjustments, when all the terms in the UPR model are equally important (see Figure 2a), the best results were obtained by the DPS and PPS with a reconfiguration period of 60 minutes ($P = 60$). In this case, our SSR solution demands many more re-computations, with almost twice the amount required by the PPS approach with $P = 5$. Specifically, the SSR mechanism re-computes the placement every two or three minutes on average. The main reason for such a high number of reconfigurations is that, unlike the DPS or PPS solutions, SSR directly depends on the instantaneous values of L_t. The latter, along with the fact that the first set of weights implies the fewest possible transformations by reassigning the affected PDU sessions, results in a UPF placement with a persistently high number of users with latency violations.



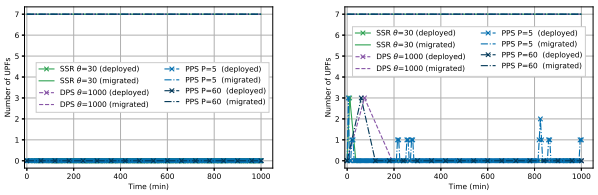
(a) Set 1 of weight factors. (b) Set 2 of weight factors.

Figure 2: Cumulative sum of reconfiguration events.



(a) Set 1 of weight factors. (b) Set 2 of weight factors.

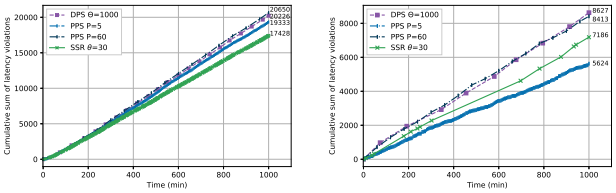
Figure 3: Cumulative sum of session relocations.



(a) Set 1 of weight factors. (b) Set 2 of weight factors.

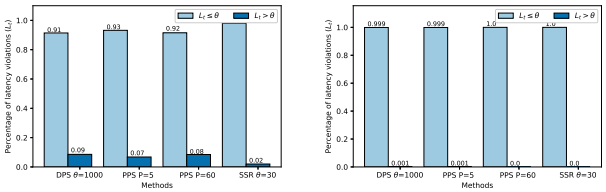
Figure 4: Number of deployed and migrated UPFs.

By contrast, when the readjustment of the placement favors the routing costs and more transformations are allowed (set 2), there is a significant reduction of more than 95% and 50% in the number of re-computations the SSR and the DPS solutions require, respectively, as shown in Figure 2b. For this set, the SSR outperformed the PPS mechanism for both reconfiguration periods, $P = 5$ and $P = 60$, with an average time of 70 minutes between reconfigurations. However, this reduction in the number of re-computations was at the expense of more session reassignments and UPF migrations, as illustrated in Figures 3 and 4. In fact, the number of reassigned sessions for the periodic mechanisms increased by more than 15 times their values w.r.t the first set of weights, despite triggering the same number of reconfiguration events. Moreover, by comparing Figures 4a and 4b, we notice variations in the UPF locations. Concretely, SSR, DPS, and PPS with $P = 60$ produced three UPF migrations, whereas the PPS with $P = 5$ had the worst performance with a total of 19 migrations. Regarding the number of deployed UPFs, this metric remained constant during the entire simulation time for all scheduling techniques in both sets of weights.



(a) Set 1 of weight factors. (b) Set 2 of weight factors.

Figure 5: Cumulative sum of latency violations.



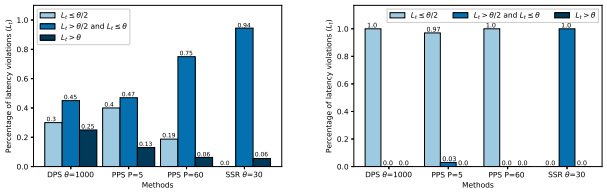
(a) Set 1 of weight factors. (b) Set 2 of weight factors.

Figure 6: Values of L_t w.r.t. θ during the simulation time.

5.3.2 Number of sessions with latency violations. In Figure 5, the total number of latency violations the system experienced for the two analyzed sets of weights is shown. In this regard, the best results were always provided via the SSR mechanism and PPS with $P = 5$. Moreover, for the first set of weights (see Figure 5a), our scheduling solution provided the highest reduction with 10% fewer latency violations than the other approaches.

Figure 6 summarizes the behavior of L_t over time w.r.t. the established QoS threshold the SSR mechanism used ($\theta = 3\%$ of users). For the first set in which the number of latency violations is high, the SSR solution provides the best results, which exceeded the threshold θ only 2% of the simulation time, as shown in Figure 6a. These results are not unexpected, since this is the main goal of the SSR mechanism. It is worth mentioning that the main reason why better performance was not obtained is that, most of the time, the threshold is exceeded almost immediately after a reconfiguration. The latter happens because there is a high probability of latency violations considering that few sessions are reassigned during the reconfigurations and that the users move at high speeds. However, when there is more stability in the placement, as a result of higher transformations (see Figure 6b), not only is the number of reconfigurations reduced, but the θ violations scarcely occur. In this case, SSR can reduce the number of latency violations above θ to zero. Thus, we can obtain results similar to those provided by solutions with frequent reconfigurations (i.e., PPS with $P = 5$).

Moreover, we analyzed the number of latency violations at the instant of the placement reevaluation. From Figures 7a and 7b, it can be seen that, unlike the other approaches, the SSR solution always triggers the reconfiguration when there is a considerable number of latency violations (i.e., $L_t > \theta/2$). Concretely, for set 1 in Figure 7a, between 18% and 40% of the placement readjustments



(a) Set 1 of weight factors.

(b) Set 2 of weight factors.

Figure 7: Values of L_t w.r.t θ at the reconfiguration moment.

the baseline solutions executed were performed when the number of sessions with latency violations was low. This situation was even worse for set 2, where between 97% and 100% of the placement readjustments were performed when $L_t \ll \theta$.

From the presented results, it has been verified that, in effect, our proposed solution for dynamic scheduling (SSR) can reduce the number of QoS violations. Specifically, in scenarios with frequent latency violations, it provides a reduction of more than 5% in the number of events with θ violations in comparison to the analyzed benchmarks. Moreover, it not only guarantees the established QoS levels but also avoids unnecessary placement recalculations, since it accounts for the instantaneous value of the selected metric, thus providing a practical and simple approach for proactive placement readjustments. Regarding periodic reconfiguration approaches, they may imply a low or high number of reevaluations according to the selected period. However, most of the time, they result in unnecessary reconfigurations or violations of the established QoS levels.

6 CONCLUSION

In this paper, we have encountered the problem of dynamic UPF placement reconfiguration as a result of user mobility. In this vein, we have proposed an ILP model to determine the optimal placement in terms of cost reduction as well as a scheduling mechanism to decide the best re-computation time. The experimental results have demonstrated that by accounting for the instantaneous values of latency violations in the system, not only the desired levels of QoS can be guaranteed, but also the number of placement reevaluations can be significantly reduced when the right reconfiguration model is selected.

Our future work includes the design of heuristic solutions to address the UPR problem as well as their evaluation in multiple scenarios. We also plan to extend the SSR mechanism by considering different metrics and investigating other stochastic optimization models, latency predictive modeling, and mobility-driven scheduling.

ACKNOWLEDGMENTS

The authors would like to extend their thanks to Richard Cziva for his valuable comments and suggestions. This work has been supported by the Ministerio de Economía y Competitividad of the Spanish Government under the project TEC2016-76795-C6-1-R and through a predoctoral FPI scholarship.

REFERENCES

[1] 3GPP. 2020. *Procedures for the 5G System (5GS); Stage 2.* Technical Specification (TS) 23.502. 3rd Generation Partnership Project (3GPP). https://www.3gpp.org/ftp/Specs/archive/23_series/23.502/23502-g40.zip Version 16.4.0.

[2] 3GPP. 2020. *System Architecture for the 5G System (5GS); Stage 2.* Technical Specification (TS) 23.501. 3rd Generation Partnership Project (3GPP). https://www.3gpp.org/ftp/Specs/archive/23_series/23.501/23501-g40.zip Version 16.4.0.

[3] 5G Americas. 2017. *5G services and use cases.* Technical Report. 5G Americas. https://www.5gamericas.org/wp-content/uploads/2019/07/5G_Service_and_Use_Cases__FINAL.pdf

[4] Christos Anagnostopoulos and Kostas Kolomvatsos. 2019. An intelligent, time-optimized monitoring scheme for edge nodes. *Journal of Network and Computer Applications* 148 (2019), 102458.

[5] Bego Blanco, Jose Oscar Fajardo, Ioannis Giannoulakis, Emmanouil Kafetzakis, Shuping Peng, Jordi Pérez-Romero, Irena Trajkovska, Pouria S Khodashenas, Leonardo Goratti, Michele Paolino, et al. 2017. Technology pillars in the architecture of future 5G mobile networks: NFV, MEC and SDN. *Computer Standards & Interfaces* 54 (2017), 216–228.

[6] Richard Cziva, Christos Anagnostopoulos, and Dimitrios P Pezaros. 2018. Dynamic, latency-optimal vNF placement at the network edge. In *IEEE INFOCOM 2018- IEEE Conference on Computer Communications*. IEEE, 693–701.

[7] Ivan Farris, Tarik Taleb, Miloud Bagaa, and Hannu Flick. 2017. Optimizing service replication for mobile delay-sensitive applications in 5G edge network. In *2017 IEEE International Conference on Communications (ICC)*. IEEE, Paris, 1–6.

[8] Thomas S Ferguson. 2006. Optimal Stopping and Applications. https://www.math.ucla.edu/~tom/Stopping/Contents.html

[9] LLC Gurobi Optimization. 2020. Gurobi Optimizer. http://www.gurobi.com

[10] William E. Hart, Carl D. Laird, Jean-Paul Watson, David L. Woodruff, Gabriel A. Hackebeil, Bethany L. Nicholson, and John D. Siirola. 2017. *Pyomo–optimization modeling in python* (second ed.). Vol. 67. Springer Science & Business Media.

[11] Jintian Hu, Gaocai Wang, Xiaotong Xu, and Yuting Lu. 2019. Study on Dynamic Service Migration Strategy with Energy Optimization in Mobile Edge Computing. *Mobile Information Systems* 2019 (2019), 1–12.

[12] Sami Kekki, Walter Featherstone, Yonggang Fang, Pekka Kuure, Alice Li, Anurag Ranjan, Debashish Purkayastha, Feng Jiangping, Danny Frydman, Gianluca Verin, Kuo-Wei Wen, Kwihoon Kim, Rohit Arora, Andy Odgers, Luis M Contreras, and Salvatore Scarpina. 2018. *MEC in 5G networks.* Technical Report. ETSI. https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp28_mec_in_5G_FINAL.pdf

[13] Irian Leyva-Pupo, Cristina Cervelló-Pastor, and Alejandro Llorens-Carrodeguas. 2019. Optimal Placement of User Plane Functions in 5G Networks. In *International Conference on Wired/Wireless Internet Communication*. Springer, 105–117.

[14] Irian Leyva-Pupo, Alejandro Santoyo-González, and Cristina Cervelló-Pastor. 2019. A Framework for the Joint Placement of Edge Service Infrastructure and User Plane Functions for 5G. *Sensors* 19, 18 (2019), 3975.

[15] Francisco J Martinez, J-C Cano, Carlos T Calafate, and Pietro Manzoni. 2008. Citymob: a mobility model pattern generator for VANETs. In *ICC Workshops-2008 IEEE International Conference on Communications Workshops*. IEEE, 370–374.

[16] Jad Oueis, Razvan Stanica, and Fabrice Valois. 2019. Virtualized Local Core Network Functions Placement in Mobile Networks. In *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 1–6.

[17] Sebastian Peters and Manzoor Ahmed Khan. 2019. Anticipatory Session Management and User Plane Function Placement for AI-Driven Beyond 5G Networks. *Procedia Computer Science* 160 (2019), 214–223.

[18] Tarik Taleb, Miloud Bagaa, and Adlen Ksentini. 2015. User mobility-aware virtual network function placement for virtual 5G network infrastructure. In *2015 IEEE International Conference on Communications (ICC)*. IEEE, 3879–3884.

[19] Hongjia Wu, Gaocai Wang, Xiaotong Xu, and Jintian Hu. 2018. A cache placement strategy for energy savings in CCN. In *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. IEEE, 788–795.